ElGamal cryptosystem-based secure authentication system for cloud-based IoT applications

Maitra, Tanmoy; Obaidat, Mohammad S.; Giri, Debasis; Dutta, Subrata; Dahal, Keshav

Published in:
IET Networks

DOI:
10.1049/iet-net.2019.0004

Published: 01/09/2019

Document Version
Peer reviewed version

IET Networks

ElGamal Cryptosystem-based Secure Authentication System for Cloud-based IoT Applications

NET-2019-0004.R1 | Research Article

Submitted on: 17-02-2019

Submitted by: Tanmoy Maitra, Mohammad Obaidat, Debasis Giri, Subrata Dutta, Keshav Dahal

Keywords: SECURITY, AUTHENTICATION, INTERNET OF THINGS

IET Research Journals

Submission Template for IET Research Journal Papers

The Institution of Engineering and Technology

ISSN 1751-8644
doi: 0000000000
www.ietdl.org

ElGamal Cryptosystem-based Secure Authentication System for Cloud-based IoT Applications

Tanmoy Maitra[1]*, *Mohammad S. Obaidat*[2], *Fellow of IEEE, Debasis Giri*[3], *Subrata Dutta*[4], *Keshav Dahal*[5]

[1] *School of Computer Engineering, KIIT, deemed to be University, Bhubaneswar, India*
[2] *M. S. Obaidat is with the Department of ECE, Nazarbayev University, Astana, Kazakhstan, KASIT, University of Jordan, Amman, Jordan, , and University of Science and Technology Bejing, China*
[3] *Department of Information Technology, Maulana Abul Kalam Azad University of Technology, Haringhata, Nadia, West Bengal*
[4] *Department of Computer Science & Engineering, National Institute of Technology, Jamshedpur, India*
[5] *School of Engineering and Computing and Artificial Intelligence, Visual Communications and Networks Research Centre, University of the West of Scotland (UWS), UK and Nanjing University of Information Science and Technology (NUIST), China*
* *E-mail: tanmoy.maitrafcs@kiit.ac.in and msobaidat@gmail.com*

Abstract: Life in modern society becomes easier due to rapid growth of different technologies like real-time analytic, ubiquitous wireless communication, commodity sensors, machine learning, and embedded systems. Nowadays, there seems to be a need to merge these technologies in the form of Internet of Things (IoT) so that smart systems can be achieved. On the other hand, cloud computing is a pillar in IoT by which end users get connected through the cloud servers for getting different services. However, to recognize the legitimacy of communicators during communication sessions through insecure channels like the Internet, serious issues in cloud based IoT applications need to be addressed. Thus authentication procedure is highly desirable to remove the unapproved access in IoT applications. This paper presents an ElGamal cryptosystem and biometric information along with a user's password-based authentication scheme for cloud based IoT applications refereed as *SAS-Cloud*. Security of the proposed scheme has been analyzed by well popular random oracle model and it is found that *SAS-Cloud* has ability to defend all the possible attacks. Furthermore, performance of *SAS-Cloud* has been evaluated and it was found that *SAS-Cloud* has better efficiency than other existing competing ElGamal cryptosystem-based authentication schemes.

1 Introduction

In modern society, connection with everyone and everything through Internet enabled electronic devices has become common for smart living [1]. To facilitate this, network research community has been trying to develop such systems so that efficient and reliable communication can be done from remote places. This networking system is known as "Internet of Things" (IoT). IoT can be stated as: it is a system, in which interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with distinctive identifiers and the ability to transfer data over a network*. There are many application areas for IoT implementation, such as *Health Care*, *Transportation*, *Industry*, *Market*, *Education*, *Vehicles*, *Smart Home* and *Agriculture*, among others. The IoT applications are developed on the top of cloud systems [2], where the cloud system acts as the enabler for the IoT applications as shown in Fig. 1. The cloud has three main features like SaaS (Software as a Service), SaaS (Storage as a service), PaaS (Platform as a Service), and IaaS (Infrastructure as a Service) [2]. Therefore, end users get different services such as medical, educational, industrial and so on, by accessing the cloud server, which is known as the service provider. In IoT applications, Internet enabled things like vehicles and sensors collect data from the area or environment and supply the data to the cloud server. The cloud server processes the data and provides a corresponding feedback. In order to get the data from remote places by accessing the server, the end users have to get permission from the server first and then agree upon a shared secret key for further secure communication within the current session which is known as remote user authentication (see Fig. 1). After getting the data, end users can also provide a feedback to the Internet enabled e-devices. A huge amount of data transaction takes place in any IoT system. For each case of data transaction, the system needs to check whether the user is authentic or not by the proper efficient and secure authentication protocol. This work concentrates on developing a secure remote user verification scheme in cloud environment of IoT applications.

* *http://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT*

- **Motivation**

There are several challenges in case of user authentication techniques. History says that no security could prove absolute secure over long period due to the smart and updated attacker. However, this study finds that most of the existing authentication schemes (can be applied in cloud based applications) do not protect systems from all security attacks. Furthermore, the existing protocols have lack of efficiency in terms of: (a) computational cost, (b) communication cost, (c) inability to detect wrong inputs during login as well as password phases, (d) extra communication overhead to alter the users' password, and (e) disclosure of the users' identity to the attacker. A proper efficient and secure authentication scheme for cloud based IoT applications should overcome or alleviate all the aforementioned issues and provide user friendly facilities.

- **Contribution**

This paper proposes a secure scheme using biometric information of users and ElGamal cryptosystem. We refer to this here as *SAS-Cloud* (Secure Authentication Scheme in Cloud based IoT Systems), to build a concrete authentication system for cloud applications. The security of *SAS-Cloud* is examined using well popular random oracle model, and the efficiency of *SAS-Cloud* is evaluated and compared with other reported competing schemes.

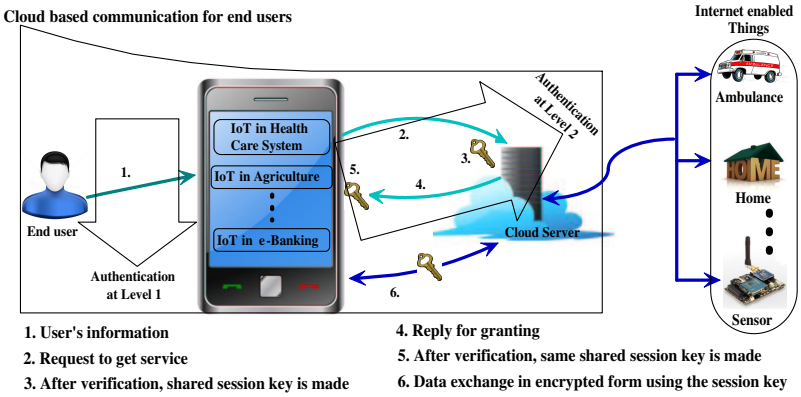

Fig. 1: A general view of Cloud based IoT architecture: authentication and secure message transmission from end users' viewpoint

The study is structured as follows. A quick overview of existing authentication protocols is highlighted in the next section. Section 3 describes some mathematical definitions, which are used in the proposed scheme. Section 4 demonstrates the adversary model as well as network model to introduce the proposed scheme. Our ElGamal-based three factor authentication scheme for cloud based IoT application, *SAS-Cloud*, is described in section 5. Security analysis of the proposed *SAS-Cloud* scheme and performance comparison of *SAS-Cloud* with related competing schemes are provided in section 6 and section 7, respectively. Advantages to used the proposed *SAS-Cloud* is given in section 8. The concluding remarks of this paper are stated in section 9.

2 Related Work

Lamport [3] first introduced a password-based authentication scheme using one way hash function. Thereafter many user authentication schemes [4–10] have been presented in this regard, which are based on only password for various Internet based applications. Jain et al. in [11] mentioned that biometric information based technology produces an effective verification tool in wireless communication. Furthermore, in many commercial, civilian, and forensic applications, biometric systems have been installed to verify identity of users [11]. Therefore, the researchers have appraised biometric with the password to amplify the degree of security [12]. Research community of this study have suggested various password and biometric based authentication schemes in [12–20]. Tan [13] presented a three-factor authentication scheme in 2013. According to Yan et al. [14], the scheme [13] is insecure from the Denial-of-Service (DoS) attack and suggested their own scheme. However, Mishra et al. [15] showed that the scheme in [14] can not protect the off-line password guessing attack and it has incompetent login, and password change phases. Chuang and Chen [16] also proposed a biometric based authentication scheme, which can be applied in cloud environment. Maitra and Giri [17] stated that an adversary can create forge message on Chuang and Chen's scheme [16], and introduced a counter measure scheme in [17]. Very recently, Wazid et al. [19] also introduced a biometric-based authentication scheme in cloud environment and after evaluating their proposed scheme through formal security analysis, the authors claimed that the proposed scheme is secure from security threats. However, the discussed authentication protocols are based on only hash function, thus the security of those schemes are dependant on hardness of one-way hash function.

On the other hand, authentication using public key cryptography like RSA based [21, 22], ElGamal based [23], Robin cryptosystem based [24], ECC-based [25–27], Bilinear Pairing based [28, 29] and so on is also well popular in the literature. However, this paper aims to design an authentication scheme for cloud based IoT application using ElGamal cryptosystem [30]. Hence we discuss only the authentication protocols using ElGamal cryptosystem reported in [23, 31–38]. Hwang and Li [23] proposed an ElGamal based authentication protocol without using any password verification table in order to eliminate password stolen attack at the server end. The authors claimed that the proposed protocol can resist different known attacks. Chan and Cheng [31] identified that different kinds of security attacks like password guessing attack, impersonation attacks, man-in the-middle attack and DoS attack can be mounted in the protocol [23]. Shen et al. [32] also proved that the protocol provided in [23] is defenceless against masquerading attack, therefore Shen et al. demonstrated a solution to prevent the masquerading attack on Hwang and Li's scheme by proposing an enhanced scheme in [32]. However, the modified protocol [32], proposed by Shen et al. is not totally secured as pointed out by Leung et al. in [33]. Yoon et al. [34] proposed a new smart card based client server authentication protocol using ElGamal signature and claimed that their protocol can resist forgery attack. However, Tian et al. in [35] argued that the protocol in [34] cannot make absolute protection against forgery attack and proposed a modified protocol in [35]. Ramasamy and Muniyandi [36] introduced a smart card based authentication protocol using ElGamal cryptosystem claiming that their protocol can withstand all the possible security threats like parallel session attack, forgery attack and denial of service attack. However Lee et al. [37] figured out that the Ramasamy and Muniyandi's scheme [36] cannot prevent all kind of attacks and they have proposed a new smart card based authentication protocol in [37] to overcome the shortcoming. Very recently, Maitra et al. [38] showed that an adversary can mount forgery attack as well as password guessing attack on Lee et al.'s scheme [37] after stealing the smart card of a legal user.

3 Preliminaries

Definition 1. *A cryptographic hash function [17, 20] can be represented as:* $\mathcal{H}: S_1 \rightarrow S_2$, *where* S_1, *a binary string of random length is taken as an input to produce a binary string* S_2 *of fixed length* l. *The cryptographic hash function* $\mathcal{H}(\cdot)$ *is said to be collision-resistant, if the following condition is maintained:*

$$Adv_{\mathcal{A}}^{\mathcal{H}}(t_1) = Pr\Big[(a_1, a_2) \in_R S_1 \times S_1 \mid (a_1 \neq a_2) \wedge \mathcal{H}(a_1) = \mathcal{H}(a_2)\Big], \quad (1)$$

where $Pr[\mathcal{E}]$ *represents the random event* $\mathcal{E}$ *produced by an adversary* $\mathcal{A}$ *for the time span* t_1 *and* $Adv_{\mathcal{A}}^{\mathcal{H}}(t_1) \leq \eta_1$, *for any small* $\eta_1 > 0$ *is the probability of advantage to find two different binary strings* a_1 *and* a_2 *over time span* t_1.

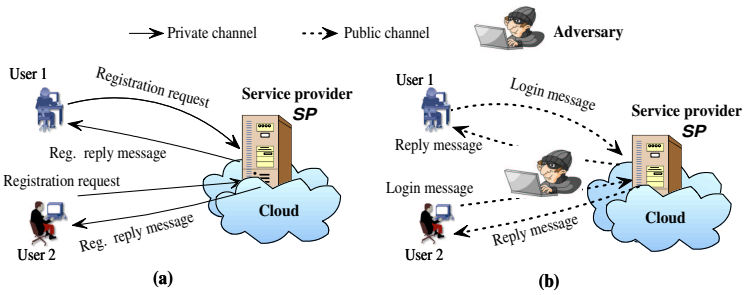

Fig. 2: Network model (a) enrollment, and (b) login and authentication

Definition 2. *A fuzzy extractor ($\mathcal{FE}$) [12] has two procedures: one is Gen, which can be represented as $Gen : MS \rightarrow \phi \times \theta$, where MS is a binary string i.e., biometric information $\mathcal{B}$ (a binary string after extracting feature of biometric by some well known mechanism like singular point extraction for fingerprint [39]), ϕ is a random string and θ is an auxiliary string, and another procedure is Rep, which can be represented as $Rep : MS \times \theta \rightarrow \phi$. The fuzzy extractor $\mathcal{FE}$ is called collision-resistant if following condition is maintained:*

$$Adv_{\mathcal{A}}^{\mathcal{FE}}(t_2) = Pr\Big[(\mathcal{B}, \mathcal{B}') \in_R MS \times MS \mid (\mathcal{B} \neq \mathcal{B}') \wedge des(\mathcal{B}, \mathcal{B}') \leq \delta d \wedge Gen(\mathcal{B}) = Gen(\mathcal{B}') \wedge Rep(\mathcal{B}, \theta) = Rep(\mathcal{B}', \theta')\Big], \quad (2)$$

where $Pr[\mathcal{E}]$ represents the random event $\mathcal{E}$ produced by an adversary $\mathcal{A}$ for the time span t_2 and $Adv_{\mathcal{A}}^{\mathcal{FE}}(t_2) \leq \eta_2$, for any small $\eta_2 > 0$ is the probability of advantage to find two different biometric strings $\mathcal{B}$ and $\mathcal{B}'$. Note that $des(\cdot)$ is a distance measurement function like Hamming distance between two different binary strings and δd is a distance tolerance value. However, both $des(\cdot)$ and δd are pre-defined in fuzzy extractor system and same for all users' biometric.

Definition 3. *Discrete Logarithm Problem (DLP) [38] states that it is hard to obtain $a \in Z_p^*$ from known inputs A, p and g such that $A = g^a \bmod p$ for any prime number p. DLP can be called a hard problem if the following condition is maintained:*

$$Adv_{\mathcal{A}}^{\mathcal{D}}(t_3) = Pr\Big[a \in Z_p^* \mid A = g^a \bmod p\Big], \quad (3)$$

where $Pr[\mathcal{E}]$ represents the random event $\mathcal{E}$ produced by an adversary $\mathcal{A}$ for the time span t_3 and $Adv_{\mathcal{A}}^{\mathcal{D}}(t_3) \leq \eta_3$, for any small $\eta_3 > 0$ is the probability of advantage to find a from given A.

4 Models

This section will discuss network and adversary models to introduce the proposed *SAS-Cloud*.

4.1 Network Model

According to the architecture of the proposed scheme, through enrollment procedure, users have to register to a service provider $\mathcal{SP}$ to get their registration conformation (See Fig. 2(a)). For this purpose, users send a request for registration to $\mathcal{SP}$ in off-line or personally. After getting the request, $\mathcal{SP}$ provides some registration information to the users so that the users can use this information in login time.

Whenever a registered user wants to get service from $\mathcal{SP}$ by accessing the mobile application via insecure channel, the user transmits a login message to $\mathcal{SP}$. Upon checking the login message, $\mathcal{SP}$ gives reply to the user. After getting the reply, the user verifies the reply message (See Fig. 2(b)). For the correct reply, both the user and $\mathcal{SP}$ agree on a secret and common session key [40].

4.2 Threat Model

This study has considered the threat model proposed by Dolev-Yao [41] to evaluate the security of the *SAS-Cloud*. According to this model [41], the communicating parties convey their message through an insecure channel during login as well as authentication phases. Therefore, an attacker $\mathcal{A}$ can capture the transmitted messages, and furthermore $\mathcal{A}$ can alter or delete the contents of the messages as shown in Fig. 2(b). The attacker $\mathcal{A}$ also acquires the information, which is stored in the user's electronic device like mobile phone, tablet or laptop by monitoring the consumption of power [42]. According to the threat model, this paper considers the following two attackers:

- *Attacks by Outsider.* A third party $\mathcal{A}$ (as an attacker), who is unrelated to this system may try to hamper in the authentication procedure by mounting various attacks. .
- *Attacks by Insider.* A valid user $\widehat{\mathcal{A}}$ (as an attacker), who is a part of the system may try to obtain confidential information of the server so that $\widehat{\mathcal{A}}$ can inject several attacks on the authentication system.

5 *SAS-Cloud*: The Proposed Scheme

In this section, we present a secure authentication scheme for IoT application using fuzzy extractor and ElGamal Cryptosystem, called as *SAS-Cloud*. Symbols and their uses are given in Table 1. *SAS-Cloud* has five phases namely, (a) set-up phase, (b) enrollment phase, (c) login phase, (d) authentication with key agreement phase and (e) password update phase.

5.1 Set-up Phase

A service provider $\mathcal{SP}$ executes algorithm $\mathscr{K}$ to get a large prime number q. $\mathcal{SP}$ picks a cyclic multiplicative group $\mathcal{G}$ of order q with a generator g. Then it picks a number s randomly such that $s \in_R Z_q^*$ and computes $PK = g^s \bmod q$. Furthermore, it selects a cryptographic hash function $\mathcal{H} : \{0,1\}^* \rightarrow \{0,1\}^k$, where k is an integer number with fixed length. Ultimately, $\mathcal{SP}$ declares public information $Param = \langle \mathcal{G}, PK, g, q, \mathcal{H}(\cdot) \rangle$ and keeps s as secret key.

5.2 Enrollment Phase

Whenever a new user $\mathcal{U}_i$ likes to enroll in the service provider $\mathcal{SP}$, registration phase is invoked using the steps shown below:

1. The user $\mathcal{U}_i$ opens the application from his/her electronic gadget and inputs his/her biometric feature (i.e., fingerprint) to a sensor

Table 1 Nomenclature used in the paper

Term	Usage
$\mathcal{G}$	A multiplicative group of prime order q
g	Generator of group $\mathcal{G}$
$\mathcal{H}(\cdot)$	Cryptographic hash function
$\mathcal{U}_i$	i-th User
$\mathcal{SP}$	Service provider
s	Secret key of $\mathcal{SP}$
PK	Public key of $\mathcal{SP}$
$\mathcal{PW}_i$	Password of $\mathcal{U}_i$
$\mathcal{ID}_i$	Identity of $\mathcal{U}_i$
$\mathcal{B}_i$	Biometric information of $\mathcal{U}_i$
r_i	Random number picked up by mobile
y_i	Random number picked up by $\mathcal{SP}$
$des(\cdot)$	Distance evaluation function
δd	Estimated difference
Y'	Parameter Y calculated or obtained by mobile
Y^*	Parameter Y obtained or calculated by $\mathcal{SP}$
SK_i	Common and secret session key between $\mathcal{U}_i$ and $\mathcal{SP}$
$\oplus$	Exclusive-OR operation
$\|$	Concatenation/append operation

enabled device like mobile. The device creates a corresponding biometric information $\mathcal{B}_i$ (defined in Definition 2 of Section 3) and provides it to $\mathcal{U}_i$.
2. $\mathcal{U}_i$ chooses an identity $\mathcal{ID}_i$, password $\mathcal{PW}_i$ and generates an unique pair (θ_i, ϕ_i) from $\mathcal{B}_i$ by computing $(\phi_i, \theta_i) \leftarrow Gen(\mathcal{B}_i)$. $\mathcal{U}_i$ then computes $\mathcal{PWR}_i = \mathcal{H}(\mathcal{PW}_i \| \phi_i)$ and sends $\langle \mathcal{ID}_i, \mathcal{PWR}_i \rangle$ to $\mathcal{SP}$ through a private channel.
3. After getting a registration request $\langle \mathcal{ID}_i, \mathcal{PWR}_i \rangle$ from $\mathcal{U}_i$, $\mathcal{SP}$ computes $A_i = \mathcal{H}(s \| \mathcal{ID}_i)$ and $D_i = \mathcal{H}(A_i)$. $\mathcal{SP}$ then checks that D_i exists in its list or not. If it is exist, $\mathcal{SP}$ gives a negative acknowledgement (i.e., decline message) to $\mathcal{U}_i$ because, received $\mathcal{ID}_i$ is not unique and it may be used by another user. In such case, $\mathcal{U}_i$ has to selects another identity until unique identity is not acquired. If D_i does not exist in its list, $\mathcal{SP}$ calculates $C_i = A_i \oplus \mathcal{PWR}_i$ and sends a registration information $\langle C_i, D_i, des(\cdot), \delta d \rangle$ to $\mathcal{U}_i$ through a private channel, where $des(\cdot)$ is a distance measurement function (defined in Definition 2 of Section 3). $\mathcal{SP}$ then updates its list $\mathcal{U}\_\mathcal{L}ist$ by incorporating D_i into it.
4. After receiving the registration information $\langle C_i, D_i, des(\cdot), \delta d \rangle$, $\mathcal{U}_i$ computes $\widehat{\mathcal{B}_i} = \mathcal{B}_i \oplus \mathcal{H}(\mathcal{ID}_i \| \mathcal{PW}_i)$, $\widehat{\theta_i} = \theta_i \oplus \mathcal{H}(\mathcal{ID}_i \| \mathcal{PW}_i)$. Finally, $\mathcal{U}_i$ stores $\langle C_i, D_i, \widehat{\mathcal{B}_i}, \widehat{\theta_i}, des(\cdot), \delta d \rangle$ into the memory of his/her electronic gadget like mobile phone. Note that, this study assumes that extracted feature from biometric i.e., binary string $\mathcal{B}_i$ and result of hash value are same bits long, which are n bits.

Fig. 3 shows the pictorial view of enrollment phase.

5.3 Login Phase

If a registered user $\mathcal{U}_i$ likes to get entry into the system by accessing the service provider $\mathcal{SP}$, login phase is invoked. $\mathcal{U}_i$ opens the application from mobile and provides his/her biometric information $\mathcal{B}_i^*$ via sensor, identity $\mathcal{ID}_i$ and password $\mathcal{PW}_i$ to the mobile. The mobile then computes the following procedures:

1. The mobile executes $\mathcal{B}_i' = \widehat{\mathcal{B}_i} \oplus \mathcal{H}(\mathcal{ID}_i \| \mathcal{PW}_i)$ and verifies $des(\mathcal{B}_i^*, \mathcal{B}_i') \leq \delta d$. If it does not satisfying the condition, $\mathcal{U}_i$ will be rejected; otherwise, it computes the next step.
2. The mobile calculates $\theta_i' = \widehat{\theta_i} \oplus \mathcal{H}(\mathcal{ID}_i \| \mathcal{PW}_i)$, $\phi_i' \leftarrow Rep(\mathcal{B}_i^*, \theta_i')$, $\mathcal{PWR}_i' = \mathcal{H}(\mathcal{PW}_i \| \phi_i')$, $A_i' = C_i \oplus \mathcal{PWR}_i'$, $D_i' = \mathcal{H}(A_i')$ and checks $D_i' =? D_i$. If equality does not preserve, the mobile refuses $\mathcal{U}_i$; otherwise, it computes the next step.
3. The mobile selects a number $r_i \in_R Z_q^*$ randomly and computes $E_i = PK^{r_i} \bmod q$, $G_i = g^{r_i} \bmod q$, $DID_i = (\mathcal{ID}_i \| r_i) \cdot E_i \bmod q$ and $F_i = \mathcal{H}(r_i \| E_i \| A_i')$. $\mathcal{U}_i$ then transmits a login request message $\langle DID_i, G_i, F_i \rangle$ to $\mathcal{SP}$ through Internet (a public channel). Note that $\mathcal{U}_i$ operates his/her mobile thus, mobile of $\mathcal{U}_i$ sends the login message on behalf of $\mathcal{U}_i$. However, in this study, we use mobile device of $\mathcal{U}_i$ and $\mathcal{U}_i$, alternatively.

5.4 Authentication with Key Agreement Phase

Upon getting the login request message $\langle DID_i, G_i, F_i \rangle$ from $\mathcal{U}_i$, $\mathcal{SP}$ computes the following steps:

1. $\mathcal{SP}$ computes $E_i^* = {G_i}^s \bmod q$, extracts $(\mathcal{ID}_i \| r_i)$ as $(\mathcal{ID}_i^* \| r_i^*) = DID_i \cdot (E_i^*)^{-1} \bmod q$ and calculates $A_i^* = \mathcal{H}(s \| \mathcal{ID}_i^*)$ and checks that D_i^* (= $\mathcal{H}(A_i^*)$) exists into $\mathcal{U}\_\mathcal{L}ist$ or not. If it does not find it, $\mathcal{SP}$ rejects $\mathcal{U}_i$; otherwise, it executes next step.

[Verification of $(\mathcal{ID}_i \| r_i) = DID_i \cdot (E_i^*)^{-1}$**]:**

$$
\begin{aligned}
& DID_i \cdot (E_i^*)^{-1} \bmod q \\
\equiv\ & (\mathcal{ID}_i \| r_i) \cdot E_i \cdot (E_i^*)^{-1}, \text{ Since } DID_i = (\mathcal{ID}_i \| r_i) \cdot E_i \\
\equiv\ & (\mathcal{ID}_i \| r_i) \cdot PK^{r_i} \cdot ({G_i}^s)^{-1}, \\
& \qquad \text{Since } E_i^* = {G_i}^s \text{ and } E_i = PK^{r_i} \\
\equiv\ & (\mathcal{ID}_i \| r_i) \cdot g^{s \cdot r_i} \cdot (g^{s \cdot r_i})^{-1}, \\
& \qquad \text{Since } PK = g^s \text{ and } G_i = g^{r_i} \\
\equiv\ & (\mathcal{ID}_i \| r_i)
\end{aligned}
$$

2. $\mathcal{SP}$ calculates $F_i^* = \mathcal{H}(r_i^* \| E_i^* \| A_i^*)$ and further checks $F_i^* =? F_i$. For the inequality, $\mathcal{SP}$ rejects the login message of $\mathcal{U}_i$; otherwise, it goes to the next step.
3. $\mathcal{SP}$ selects a number $y_i \in_R Z_q^*$ random and further computes $Q_i = A_i^* \oplus y_i$, $SK_i = \mathcal{H}(y_i \| r_i^*)$, $L_i = \mathcal{H}(\mathcal{ID}_i^* \| SK_i \| A_i^*)$ and sends a reply message $\langle Q_i, L_i \rangle$ to $\mathcal{U}_i$ via a insecure channel. $\mathcal{SP}$ accepts SK_i as a common and secret session key.

After getting the reply message $\langle Q_i, L_i \rangle$ from $\mathcal{SP}$, the mobile of $\mathcal{U}_i$ performs the following step to authenticate the reply message of $\mathcal{SP}$:

1. The mobile calculates $y_i' = A_i' \oplus Q_i$, $SK_i' = \mathcal{H}(y_i' \| r_i)$, $L_i' = \mathcal{H}(\mathcal{ID}_i \| SK_i' \| A_i')$ and checks $L_i' =? L_i$. If the equality is satisfied, $\mathcal{U}_i$ concurs upon the common secret key SK_i' (= SK_i); otherwise, it refuses the reply message.

Fig. 4 depicts a pictorial view of login and authentication with key agreement phases.

5.5 Password Update Phase

If a user $\mathcal{U}_i$ likes to alter his/her password, this phase is invoked. $\mathcal{U}_i$ opens the application from mobile and provides his/her biometric information $\mathcal{B}_i^*$ through sensor, identity $\mathcal{ID}_i$ and password $\mathcal{PW}_i$ to the mobile. The mobile then executes the following steps:

$\mathcal{U}_i$		$\mathcal{SP}$
Extract biometric $\mathcal{B}_i$ from sensor		
Choose $\mathcal{ID}_i$ and $\mathcal{PW}_i$		
Generate $(\phi_i, \theta_i) \leftarrow Gen(\mathcal{B}_i)$		
Compute $\mathcal{PWR}_i = \mathcal{H}(\mathcal{PW}_i \| \phi_i)$	Supply $\langle \mathcal{ID}_i, \mathcal{PWR}_i \rangle \rightarrow$	Compute $A_i = \mathcal{H}(s \| \mathcal{ID}_i)$, $D_i = \mathcal{H}(A_i)$
		Check D_i is present in $\mathcal{U}\_\mathcal{L}ist$
		For availability, compute $C_i = A_i \oplus \mathcal{PWR}_i$
Compute $\widehat{\mathcal{B}_i} = \mathcal{B}_i \oplus \mathcal{H}(\mathcal{ID}_i \| \mathcal{PW}_i)$	Provide $\langle C_i, D_i, des(\cdot), \delta d \rangle \leftarrow$	Add D_i into $\mathcal{U}\_\mathcal{L}ist$
$\widehat{\theta_i} = \theta_i \oplus \mathcal{H}(\mathcal{ID}_i \| \mathcal{PW}_i)$		
Store $\langle C_i, D_i, des(\cdot), \widehat{\mathcal{B}_i}, \widehat{\theta_i}, \delta d \rangle$ into mobile		

Fig. 3: Enrollment phase of *SAS-Cloud*

$\mathcal{U}_i$	Public channel	$\mathcal{SP}$
Submit $\mathcal{B}_i^*$, $\mathcal{ID}_i$ and $\mathcal{PW}_i$		
Compute $\mathcal{B}_i' = \widehat{\mathcal{B}_i} \oplus \mathcal{H}(\mathcal{ID}_i \| \mathcal{PW}_i)$		
Check $des(\mathcal{B}_i^*, \mathcal{B}_i') \leq \delta d$		
Compute $\theta_i' = \widehat{\theta_i} \oplus \mathcal{H}(\mathcal{ID}_i \| \mathcal{PW}_i)$		
$\phi_i' \leftarrow Rep(\mathcal{B}_i^*, \theta_i')$, $\mathcal{PWR}_i' = \mathcal{H}(\mathcal{PW}_i \| \phi_i')$		
$A_i' = C_i \oplus \mathcal{PWR}_i'$, $D_i' = \mathcal{H}(A_i')$		
Verify $D_i' =? D_i$		
Pick $r_i \in_R Z_q^*$		
Compute $E_i = PK^{r_i} \bmod q$		
$G_i = g^{r_i} \bmod q$, $F_i = \mathcal{H}(r_i \| E_i \| A_i')$		
$DID_i = (\mathcal{ID}_i \| r_i) \cdot E_i \bmod q$	$\langle DID_i, G_i, F_i \rangle \rightarrow$	Compute $E_i^* = G_i{}^s \bmod q$
		Extract $(\mathcal{ID}_i^* \| r_i^*) = DID_i \cdot E_i^{*-1} \bmod q$
		Calculate $A_i^* = \mathcal{H}(s \| \mathcal{ID}_i^*)$
		Check $\mathcal{H}(A_i^*)$ is present in $\mathcal{U}\_\mathcal{L}ist$
		Compute $F_i^* = \mathcal{H}(r_i^* \| E_i^* \| A_i^*)$
		Check $F_i^* =? F_i$
		Choose $y_i \in_R Z_q^*$
		Compute $Q_i = A_i^* \oplus y_i$, $SK_i = \mathcal{H}(y_i \| r_i^*)$
		$L_i = \mathcal{H}(\mathcal{ID}_i^* \| SK_i \| A_i^*)$
Compute $y_i' = A_i' \oplus Q_i$	$\langle Q_i, L_i \rangle \leftarrow$	SK_i as common secret session key
$SK_i' = \mathcal{H}(y_i' \| r_i)$, $L_i' = \mathcal{H}(\mathcal{ID}_i \| SK_i' \| A_i')$		
Check $L_i' =? L_i$		
SK_i' (= SK_i) as shared secret session key		

Fig. 4: Login and authentication with key agreement phases of SAS-Cloud

1. The mobile calculates $\mathcal{B}_i' = \widehat{\mathcal{B}_i} \oplus \mathcal{H}(\mathcal{ID}_i \| \mathcal{PW}_i)$ and verifies $des(\mathcal{B}_i^*, \mathcal{B}_i') \leq \delta d$. For not satisfying the condition, $\mathcal{U}_i$ will be rejected; otherwise, it computes the next step.
2. The mobile calculates $\theta_i' = \widehat{\theta_i} \oplus \mathcal{H}(\mathcal{ID}_i \| \mathcal{PW}_i)$, $\phi_i' \leftarrow Rep(\mathcal{B}_i^*, \theta_i')$, $\mathcal{PWR}_i' = \mathcal{H}(\mathcal{PW}_i \| \phi_i')$, $A_i' = C_i \oplus \mathcal{PWR}_i'$, $D_i' = \mathcal{H}(A_i')$ and verifies $D_i' =? D_i$. If equality does not hold, $\mathcal{U}_i$ will be rejected; otherwise, the mobile device is allowed to enter a new password.
3. $\mathcal{U}_i$ picks a new password $\mathcal{PW}_i^{[new]}$ and inputs it to the mobile. The mobile then executes $\mathcal{PWR}_i^{[new]} = \mathcal{H}(\mathcal{PW}_i^{[new]} \| \phi_i')$, $C_i^{[new]} = A_i' \oplus \mathcal{PWR}_i^{[new]}$, $\widehat{\mathcal{B}}_i^{[new]} = \mathcal{B}_i' \oplus \mathcal{H}(\mathcal{ID}_i \| \mathcal{PW}_i^{[new]})$ and $\widehat{\theta}_i^{[new]} = \theta_i' \oplus \mathcal{H}(\mathcal{ID}_i \| \mathcal{PW}_i^{[new]})$. The mobile then replaces C_i, $\widehat{\mathcal{B}_i}$ and $\widehat{\theta_i}$ with $C_i^{[new]}$, $\widehat{\mathcal{B}}_i^{[new]}$ and $\widehat{\theta}_i^{[new]}$, respectively.

6 Security Analysis of *SAS-Cloud*

This study has done the formal security analysis of *SAS-Cloud* under the random oracle model. We define the random oracles for the formal security analysis of *SAS-Cloud* as follows:

- A random oracle $\mathcal{O}racle\mathcal{H}$ keeps a tuple $\langle u, v \rangle$ such that $v = \mathcal{H}(u)$. It supplies u from v upon getting a query $(q\mathcal{H}, v)$ if $\langle u, v \rangle$ is exist in the tuple; otherwise, produces a number r_1 randomly. Then it reserves $\langle r_1, v \rangle$ into its tuple as a new entry.
- Random oracle $\mathcal{O}racle\mathcal{FE}$ contains two parts:
1. From a tuple $\langle \mathcal{B}, \phi, \theta \rangle$, $\mathcal{O}racle\mathcal{FE}_{Gen}$ unconditionally generates the pair (ϕ, θ) after getting a query $(qGen, \mathcal{B})$ such that $(\phi, \theta) \leftarrow Gen(\mathcal{B})$ if $\langle \mathcal{B}, \phi, \theta \rangle$ exists in its tuple; else, it supplies two numbers r_2 and r_3 randomly. Then it reserves a new entry $\langle \mathcal{B}, r_2, r_3 \rangle$ into its tuple.
2. From a tuple $\langle \mathcal{B}', \phi, \theta \rangle$, $\mathcal{O}racle\mathcal{FE}_{Rep}$ unconditionally generates ϕ after getting a query $(qRep, \mathcal{B}', \theta)$ such that $\phi \leftarrow Rep(\mathcal{B}', \theta)$ if $\langle \mathcal{B}', \phi, \theta \rangle$ exists in its tuple; else, it supplies a number r_4 randomly. Then it reserves $\langle \mathcal{B}', r_4, \theta \rangle$ into its tuple as a new entry .
- Random oracle $\mathcal{O}racle\mathcal{D}$ keeps a tuple $\langle c_1, c_2, g, q \rangle$ such that $c_2 = g^{c_1} \bmod q$. It produces c_1 from c_2 upon getting a query $(q\mathcal{D}, c_2)$ if $\langle c_1, c_2 \rangle$ exists in its tuple; otherwise, supplies a number r_5 randomly. Then it reserves $\langle r_5, c_2, g, q \rangle$ into its tuple as a new entry .

Theorem 1. *Under the assumption that a fuzzy extractor $\mathcal{FE}$ and cryptographic hash function $\mathcal{H}(\cdot)$ represent the random oracles,* SAS-Cloud *is provably secure against an attacker $\mathcal{A}$ for acquiring the password $\mathcal{PW}_i$, biometric $\mathcal{B}_i$ and identity $\mathcal{ID}_i$ of a user $\mathcal{U}_i$ even if $\mathcal{A}$ obtains the parameters that are preserved into the mobile of $\mathcal{U}_i$ and captures the communication messages between $\mathcal{U}_i$ and the service provider $\mathcal{SP}$.*

Algorithm 1 $EXP1^{oracle}_{\mathcal{A},\ SAS-Cloud}$

Input: $C_i, D_i, \widehat{\mathcal{B}}_i, \widehat{\theta}_i, DID_i, G_i, F_i, L_i, PK, g, q$

Output: 0 or 1; 0: **Fail** and 1: **Win**

Asks $\mathcal{O}racle\mathcal{H}$ on the input D_i to obtain the information A_i (= $\mathcal{H}(s\|\mathcal{ID}_i)$) as $(A_i^*) \leftarrow \mathcal{O}racle\mathcal{H}(D_i)$
Asks $\mathcal{O}racle\mathcal{H}$ on the input F_i to get the information A_i, r_i and E_i as $(r_i^*\|E_i^*\|A_i^{**}) \leftarrow \mathcal{O}racle\mathcal{H}(F_i)$
Asks $\mathcal{O}racle\mathcal{H}$ on the input L_i to get the information $SK_i, \mathcal{ID}_i$ and A_i as $(\mathcal{ID}_i^*\|SK_i^*\|A_i^{***}) \leftarrow \mathcal{O}racle\mathcal{H}(L_i)$
Computes $E_i^{**} = PK^{r_i^*} \bmod q$ and $G_i^* = g^{r_i^*} \bmod q$
if $(A_i^{***} == A_i^{**} == A_i^*)$ && $(E_i^* == E_i^{**})$ && $(G_i == G_i^*)$ **then**
  Computes $(\mathcal{ID}_i^{**}\|r_i^{**}) = DID_i \cdot (E_i^*)^{-1} \bmod q$
  Asks $\mathcal{O}racle\mathcal{H}$ on the input A_i^* to get the information s and $\mathcal{ID}_i$)) as $(s^*\|\mathcal{ID}_i^{***}) \leftarrow \mathcal{O}racle\mathcal{H}(A_i^*)$
  if $(r_i^{**} == r_i^*)$ && $(\mathcal{ID}_i^{**} == \mathcal{ID}_i^* == \mathcal{ID}_i^{***})$ **then**
    Computes $\mathcal{PWR}_i^* = C_i \oplus A_i^*$
    repeat
      Chooses a password $\mathcal{PW}_i^{[guess]}$
      Computes $\mathcal{B}_i^* = \widehat{\mathcal{B}}_i \oplus \mathcal{H}(\mathcal{ID}_i^*\|\mathcal{PW}_i^{[guess]})$ and $\theta_i^* = \widehat{\theta}_i \oplus \mathcal{H}(\mathcal{ID}_i^*\|\mathcal{PW}_i^{[guess]})$
      Asks $\mathcal{O}racle\mathcal{FE}_{Rep}$ on the input $\mathcal{B}_i^*$ and θ_i^* to get the information ϕ_i, as $(\phi_i^*) \leftarrow \mathcal{O}racle\mathcal{FE}_{Rep}(\mathcal{B}_i^*, \theta_i^*)$
      Calculates $\mathcal{PWR}_i^{[guess]} = \mathcal{H}(\mathcal{PW}_i^{[guess]}\|\phi_i^*)$
    until $(\mathcal{PWR}_i^{[guess]} == \mathcal{PWR}_i^*)$
    if $(\mathcal{PWR}_i^{[guess]} == \mathcal{PWR}_i^*)$ **then**
      Accepts $\mathcal{PW}_i^{[guess]}$, $\mathcal{ID}_i^*$ and $\mathcal{B}_i^*$ as correct password, identity and biometric
      Return 1
    else
      Return 0
    end if
  else
    Return 0
  end if
else
  Return 0
end if

Proof: This study constructs an attacker $\mathcal{A}$ who has the ability to obtain the password $\mathcal{PW}_i$, identity $\mathcal{ID}_i$ and biometric information $\mathcal{B}_i$ of $\mathcal{U}_i$. In this regards, this work assumes that the mobile device of a user $\mathcal{U}_i$ is lost or stolen. Therefore, $\mathcal{A}$ can obtain the stored information $\langle C_i, D_i, \widehat{\mathcal{B}_i}, \widehat{\theta_i}\rangle$ from the memory of mobile of $\mathcal{U}_i$ by calculating power consumption [42]. The attacker $\mathcal{A}$ also captures the login request message $\langle DID_i, G_i, F_i\rangle$ and a reply message $\langle Q_i, L_i\rangle$. The adversary $\mathcal{A}$ executes the experiment, $EXP1^{oracle}_{\mathcal{A},\ SAS-Cloud}$ for our secure authentication scheme (SAS-Cloud) to get the password $\mathcal{PW}_i$, identity $\mathcal{ID}_i$ and biometric parameter $\mathcal{B}_i$ of the user $\mathcal{U}_i$ as discussed in Algorithm 1.

This study defines the success probability for $EXP1^{oracle}_{\mathcal{A},\ SAS-Cloud}$ as $Succ1^{oracle}_{\mathcal{A},\ SAS-Cloud} = |2Pr[EXP1^{oracle}_{\mathcal{A},\ SAS-Cloud} = 1] - 1|$. Then the advantage of $EXP1^{oracle}_{\mathcal{A},\ SAS-Cloud}$ is given by $Adv1^{oracle}_{\mathcal{A},\ SAS-Cloud}(t, q\mathcal{H}, q\mathcal{FE}) = max_{\mathcal{A}}\{Succ1^{oracle}_{\mathcal{A},\ SAS-Cloud}\}$, where the maximum is considered over all $\mathcal{A}$ with the implementation time t, $q\mathcal{H}$ and $q\mathcal{FE}$ are the # of queries submitted to $\mathcal{O}racle\mathcal{H}$ and $\mathcal{O}racle\mathcal{FE}$ oracles, respectively. It can be said that *SAS-Cloud* is provably secure against the attacker $\mathcal{A}$ for obtaining the password $\mathcal{PW}_i$, identity $\mathcal{ID}_i$ and biometric information $\mathcal{B}_i$ of $\mathcal{U}_i$, if $Adv1^{oracle}_{\mathcal{A},\ SAS-Cloud}(t, q\mathcal{H}, q\mathcal{FE}) \leq \eta$, for any negligible $\eta > 0$. $EXP1^{oracle}_{\mathcal{A},\ SAS-Cloud}$ (see Algorithm 1) shows that, if $\mathcal{A}$ earns success to execute the reverse of cryptographic hash function $\mathcal{H}(\cdot)$ and can explore the hardness of fuzzy extractor, then $\mathcal{A}$ will correctly obtain the password $\mathcal{PW}_i$, identity $\mathcal{ID}_i$ and biometric parameter $\mathcal{B}_i$ of $\mathcal{U}_i$ by employing random oracles $\mathcal{O}racle\mathcal{H}$ and $\mathcal{O}racle\mathcal{FE}$, respectively, and secures the win in this game. However, according to Definitions 1 and 2, we can write that $Adv_{\mathcal{A}}^{\mathcal{O}racle\mathcal{H}}(t) \leq \eta_1$, for any negligible $\eta_1 > 0$ and $Adv_{\mathcal{A}}^{\mathcal{O}racle\mathcal{FE}}(t) \leq \eta_2$, for any negligible $\eta_2 > 0$. Therefore, we get $Adv1^{oracle}_{\mathcal{A},\ SAS-Cloud}(t, q\mathcal{H}, q\mathcal{FE}) \leq \eta$, for any negligible $\eta > 0$ as the *SAS-Cloud* depends on both $Adv_{\mathcal{A}}^{\mathcal{O}racle\mathcal{H}}(t)$ and $Adv_{\mathcal{A}}^{\mathcal{O}racle\mathcal{FE}}(t)$. Therefore, *SAS-Cloud* provides the security against the attacker $\mathcal{A}$ for obtaining the password $\mathcal{PW}_i$, identity $\mathcal{ID}_i$ and biometric information $\mathcal{B}_i$ of $\mathcal{U}_i$. □

Theorem 2. *Under the assumption that DLP and cryptographic hash function $\mathcal{H}(\cdot)$ represent the random oracles,* SAS-Cloud *is provably secure against an attacker $\mathcal{A}$ for getting the private key s of service provider $\mathcal{SP}$ even if $\mathcal{A}$ knows the parameters that are reserved into U_i's mobile and captures the communication messages between $\mathcal{U}_i$ and $\mathcal{SP}$.*

Proof: This work constructs an attacker $\mathcal{A}$ who has the ability to get the private key s of the service provider $\mathcal{SP}$. However, we consider the same suppositions discussed in Theorem 1. The attacker $\mathcal{A}$ executes the experiment, $EXP2^{oracle}_{\mathcal{A},\ SAS-Cloud}$ for the secure authentication scheme (*SAS-Cloud*) to get the private key s of the service provider $\mathcal{SP}$ as provided in Algorithm 2.

We define the success probability for $EXP2^{oracle}_{\mathcal{A},\ SAS-Cloud}$ as $Succ2^{oracle}_{\mathcal{A},\ SAS-Cloud} = |2Pr[EXP2^{oracle}_{\mathcal{A},\ SAS-Cloud} = 1] - 1|$. Then the advantage of $EXP2^{oracle}_{\mathcal{A},\ SAS-Cloud}$ is given by $Adv2^{oracle}_{\mathcal{A},\ SAS-Cloud}(t, q\mathcal{H}, q\mathcal{D}) = max_{\mathcal{A}}\{Succ2^{oracle}_{\mathcal{A},\ SAS-Cloud}\}$, where we have considered the maximum over all $\mathcal{A}$ with the implementation time t, # of queries $q\mathcal{H}$, $q\mathcal{D}$ submitted to $\mathcal{O}racle\mathcal{H}$ and $\mathcal{O}racle\mathcal{D}$ oracles, respectively. The proposed *SAS-Cloud* is provably secure against the attacker $\mathcal{A}$ for obtaining the secret key s of the service provider $\mathcal{SP}$, if $Adv2^{oracle}_{\mathcal{A},\ SAS-Cloud}(t, q\mathcal{H}) \leq \eta_1$, for

Algorithm 2 $EXP2^{oracle}_{\mathcal{A},\ SAS-Cloud}$

Input: $D_i, DID_i, G_i, F_i, L_i, PK, g, q, Q_i$

Output: 0 or 1; 0: **Fail**, 1: **Win**

Asks $\mathcal{O}racle\mathcal{H}$ on the input D_i to obtain the information A_i (= $\mathcal{H}(s\|\mathcal{ID}_i)$) as $(A_i^*) \leftarrow \mathcal{O}racle\mathcal{H}(D_i)$
Asks $\mathcal{O}racle\mathcal{H}$ on the input F_i to get the information A_i, r_i and E_i as $(r_i^*\|E_i^*\|A_i^{**}) \leftarrow \mathcal{O}racle\mathcal{H}(F_i)$
Asks $\mathcal{O}racle\mathcal{D}$ on the input PK (= $g^s \bmod q$) to obtain the information s as $(s^*) \leftarrow \mathcal{O}racle\mathcal{D}(PK)$
Asks $\mathcal{O}racle\mathcal{H}$ on the input L_i to get the information $SK_i, \mathcal{ID}_i$ and A_i as $(\mathcal{ID}_i^*\|SK_i^*\|A_i^{***}) \leftarrow \mathcal{O}racle\mathcal{H}(L_i)$
Asks $\mathcal{O}racle\mathcal{D}$ on the input G_i (= $g^{r_i} \bmod q$) and g to get the information r_i as $(r_i^{**}) \leftarrow \mathcal{O}racle\mathcal{D}(G_i, g)$
if $(r_i^* == r_i^{**})$ && $(A_i^{***} == A_i^{**} == A_i^*)$ **then**
  Computes $E_i^{**} = PK^{r_i^*} \bmod q$
  if $(E_i^* == E_i^{**})$ **then**
    Computes $(\mathcal{ID}_i^{**}\|r_i^{***}) = DID_i \cdot (E_i^*)^{-1} \bmod q$
    Asks $\mathcal{O}racle\mathcal{H}$ on the input A_i^* to get the information s and $\mathcal{ID}_i$)) as $(s^{**}\|\mathcal{ID}_i^{***}) \leftarrow \mathcal{O}racle\mathcal{H}(A_i^*)$
    if $(r_i^{***} == r_i^*)$ && $(\mathcal{ID}_i^{**} == \mathcal{ID}_i^* == \mathcal{ID}_i^{***})$ && $(s^* == s^{**})$ **then**
      Accepts s^* as private key of $\mathcal{SP}$
      Return 1
    else
      Return 0
    end if
  else
    Return 0
  end if
else
  Return 0
end if

any negligible $\eta_1 > 0$ and $Adv2^{oracle}_{\mathcal{A},\ SAS-Cloud}(t, q\mathcal{D}) \leq \eta_2$, for any small $\eta_2 > 0$. According to algorithm $EXP2^{oracle}_{\mathcal{A},\ SAS-Cloud}$ (see Algorithm 2), if the attacker $\mathcal{A}$ earns success to execute the inversion of cryptographic hash function $\mathcal{H}(\cdot)$ as well as earns success to crack DLP, then $\mathcal{A}$ can correctly obtain the secret key s by employing random oracles $\mathcal{O}racle\mathcal{H}$ and $\mathcal{O}racle\mathcal{D}$, respectively and secures the win in this game. However, according to Definition 1 and Definition 3, it can be written that $Adv_{\mathcal{A}}^{\mathcal{O}racle\mathcal{H}}(t) \leq \eta_1$, for any negligible $\eta_1 > 0$ and $Adv_{\mathcal{A}}^{\mathcal{O}racle\mathcal{D}}(t) \leq \eta_2$, for any negligible $\eta_2 > 0$. Since, we get $Adv2^{oracle}_{\mathcal{A},\ SAS-Cloud}(t, q\mathcal{H}, q\mathcal{D}) \leq \eta$, for any negligible $\eta > 0$ because, *SAS-Cloud* depends on $Adv_{\mathcal{A}}^{\mathcal{O}racle\mathcal{H}}(t)$ as well as $Adv_{\mathcal{A}}^{\mathcal{O}racle\mathcal{D}}(t)$. Thus, the proposed *SAS-Cloud* is providing the security against the attacker $\mathcal{A}$ for obtaining s of the service provider $\mathcal{SP}$. □

Theorem 3. *Under the assumption that the DLP and cryptographic hash function $\mathcal{H}(\cdot)$ represent the random oracles,* SAS-Cloud *is provably secure against an attacker $\mathcal{A}$ for obtaining the common secret session key SK_i between $\mathcal{U}_i$ and $\mathcal{SP}$ even if $\mathcal{A}$ knows the parameters that are reserved into U_i's mobile device and captures the communication messages between $\mathcal{U}_i$ and $\mathcal{SP}$.*

Proof: This work constructs an attacker $\mathcal{A}$ who has ability to obtain the session key SK_i between a user and the service provider $\mathcal{SP}$. However, we consider the same suppositions discussed in Theorem 1. The attacker $\mathcal{A}$ executes the experiment, $EXP3^{oracle}_{\mathcal{A},\ SAS-Cloud}$ for the secure authentication scheme (*SAS-Cloud*) to obtain the session key SK_i between $\mathcal{U}_i$ and $\mathcal{SP}$ as given in Algorithm 3.

We define the success probability of $EXP3^{oracle}_{\mathcal{A},\ SAS-Cloud}$ as $Succ3^{oracle}_{\mathcal{A},\ SAS-Cloud} = |2Pr[EXP3^{oracle}_{\mathcal{A},\ SAS-Cloud} = 1] - 1|$. Then the advantage of $EXP3^{oracle}_{\mathcal{A},\ SAS-Cloud}$ is given by $Adv3^{oracle}_{\mathcal{A},\ SAS-Cloud}(t, q\mathcal{H}, q\mathcal{D}) = max_{\mathcal{A}}\{Succ3^{oracle}_{\mathcal{A},\ SAS-Cloud}\}$, where we have taken the maximum over all $\mathcal{A}$ with the implementation time t, $q\mathcal{H}$ and $q\mathcal{D}$ are the # of queries asked to $\mathcal{O}racle\mathcal{H}$ and $\mathcal{O}racle\mathcal{D}$ oracles, respectively. We can say *SAS-Cloud* is provably secure against the attacker $\mathcal{A}$ for obtaining the session key SK_i between $\mathcal{U}_i$ and the service provider $\mathcal{SP}$, if $Adv3^{oracle}_{\mathcal{A},\ SAS-Cloud}(t, q\mathcal{H}) \leq \eta_1$, for any negligible $\eta_1 > 0$ and $Adv3^{oracle}_{\mathcal{A},\ SAS-Cloud}(t, q\mathcal{D}) \leq \eta_2$, for any negligible $\eta_2 > 0$. $EXP3^{oracle}_{\mathcal{A},\ SAS-Cloud}$ (discussed in Algorithm 3) shows, if the attacker $\mathcal{A}$ is able to calculate reverse of the cryptographic hash function $\mathcal{H}(\cdot)$ and also cracks DLP, then $\mathcal{A}$ can get success to derive the session key SK_i by employing the $\mathcal{O}racle\mathcal{H}$ and $\mathcal{O}racle\mathcal{D}$ random oracles, and gets victory in the game. Nevertheless, after observing Definition 1 and Definition 3, we can write $Adv_{\mathcal{A}}^{\mathcal{O}racle\mathcal{H}}(t) \leq \eta_1$, for any negligible $\eta_1 > 0$ and $Adv_{\mathcal{A}}^{\mathcal{O}racle\mathcal{D}}(t) \leq \eta_2$, for any negligible $\eta_2 > 0$. Since, we obtain $Adv3^{oracle}_{\mathcal{A},\ SAS-Cloud}(t, q\mathcal{H}, q\mathcal{D}) \leq \eta$, for any negligible $\eta > 0$ as, *SAS-Cloud* depends on $Adv_{\mathcal{A}}^{\mathcal{O}racle\mathcal{H}}(t)$ as well as $Adv_{\mathcal{A}}^{\mathcal{O}racle\mathcal{D}}(t)$. Thus, the proposed *SAS-Cloud* is providing security against $\mathcal{A}$ for obtaining SK_i between the user $\mathcal{U}_i$ and the service provider $\mathcal{SP}$. □

Theorem 4. *A registered user $\mathcal{U}_{\widehat{\mathcal{A}}}$ as an attacker cannot extract the private key s of the service provider $\mathcal{SP}$ even if he/she has stored parameters into his/her mobile device.*

Proof: A registered user say, $\mathcal{U}_i$ as an attacker $\widehat{\mathcal{A}}$ may try to login into *SAS-Cloud* as an another valid user say, $\mathcal{U}_j$. To do so, $\widehat{\mathcal{A}}$ must knows the private key s of the service provider $\mathcal{SP}$. Since, $\widehat{\mathcal{A}}$ is a legal user, $\widehat{\mathcal{A}}$ knows his/her identity $\mathcal{ID}_i$, password $\mathcal{PW}_i$ and biometric parameter ϕ_i. Therefore, $\widehat{\mathcal{A}}$ is able to calculate $\mathcal{H}(s\|\mathcal{ID}_i)$ by executing $C_i \oplus \mathcal{H}(\mathcal{PW}_i\|\phi_i)$, where C_i is the stored information into his/her mobile device. However, from $\mathcal{H}(s\|\mathcal{ID}_i)$, $\widehat{\mathcal{A}}$ unable to derive s due to hardness of the reverse of cryptographic hash function. In addition, Theorem 2 exhibits that s cannot be derived from familiar parameters. As a result, $\widehat{\mathcal{A}}$ cannot produce any security attacks on *SAS-Cloud*.

□

Algorithm 3 $EXP3^{oracle}_{\mathcal{A},\ SAS-Cloud}$

Input: $D_i, DID_i, G_i, F_i, L_i, PK, g, q, Q_i$

Output: 0 or 1; 0: **Fail**, 1: **Win**

Asks $\mathcal{O}racle\mathcal{H}$ on the input D_i to obtain the information A_i (= $\mathcal{H}(s\|\mathcal{ID}_i)$) as $(A_i^*) \leftarrow \mathcal{O}racle\mathcal{H}(D_i)$
Asks $\mathcal{O}racle\mathcal{H}$ on the input F_i to get the information A_i, r_i and E_i as $(r_i^*\|E_i^*\|A_i^{**}) \leftarrow \mathcal{O}racle\mathcal{H}(F_i)$
Asks $\mathcal{O}racle\mathcal{D}$ on the input PK (= g^s mod q) to retrieve the information s as $(s^*) \leftarrow \mathcal{O}racle\mathcal{D}(PK)$
Asks $\mathcal{O}racle\mathcal{H}$ on the input L_i to get the information SK_i, $\mathcal{ID}_i$ and A_i as $(\mathcal{ID}_i^*\|SK_i^*\|A_i^{***}) \leftarrow \mathcal{O}racle\mathcal{H}(L_i)$
Asks $\mathcal{O}racle\mathcal{D}$ on the input G_i (= g^{r_i} mod q) to retrieve the information r_i as $(r_i^{**}) \leftarrow \mathcal{O}racle\mathcal{D}(G_i)$
if $(A_i^{***} == A_i^{**} == A_i^*)$ && $(r_i^* == r_i^{**})$ **then**
  Computes $E_i^{**} = PK^{r_i^*}$ mod q and $y_i^* = Q_i \oplus A_i^*$
  if $(E_i^* == E_i^{**})$ **then**
    Computes $(\mathcal{ID}_i^{**}\|r_i^{***}) = DID_i \cdot (E_i^*)^{-1}$ mod q
    if $(r_i^{***} == r_i^*)$ && $(\mathcal{ID}_i^{**} == \mathcal{ID}_i^*)$ **then**
      Executes $SK_i^{**} = \mathcal{H}(y_i^*\|r_i^*)$
      if $(SK_i^* == SK_i^{**})$ **then**
        SK_i^* is accepted as the common session key
        Return 1
      else
        Return 0
      end if
    else
      Return 0
    end if
  else
    Return 0
  end if
else
  Return 0
end if

6.1 Remarks on Proposed Theorems

Theorem 1 demonstrates that *SAS-Cloud* is providing security against the *off-line password guessing attack*.

In *SAS-Cloud*, $\mathcal{A}$ cannot produce the *forgery attack* without knowing the $\mathcal{PW}_i$ and biometric information $\mathcal{B}_i$ of a user $\mathcal{U}_i$ and the private key s of the service provider $\mathcal{SP}$. Theorem 1 and Theorem 2 show that the confidential parameters of the service provider and the user are well protected from the attacker. As a result, there is no feasibility to produce the *forgery attack* on *SAS-Cloud*.

Furthermore, Theorem 3 shows that *SAS-Cloud* can protect the *session key obtaining attack* because, without any knowledge of random nonce(s) r_i and y_i, $\mathcal{A}$ cannot derive the session key SK_i.

In *SAS-Cloud*, the communicating messages are computed using random numbers. Hence, the messages are assuring to be non-identical for each session. As a result, $\mathcal{A}$ cannot create the *replay attack* on *SAS-Cloud*. In addition, Denial of Service (DoS) attack is easily identified in *SAS-Cloud* (See Section 8).

7 Performance Evaluation

Here, the performances of *SAS-Cloud* are compared with the competing existing authentication schemes namely, Tan's scheme [13], Yan et al.'s scheme [14], Mishra et al.'s scheme [15], Chuang and Chen's scheme [16], Hwang and Li's scheme [23], Shen et al.'s scheme [32], Yoon et al.'s scheme [34], Ramasamy and Muniyandi's scheme [36] and Lee et al.'s scheme [37]. The compared schemes in [13–16, 23, 32, 34, 36, 37] are not usable for practical scenarios because, these schemes are not resisting the security attacks (See Table 2). In the related work section of this work, we have described that most of the proposed schemes are insecure against security attacks. Furthermore, analysis of security of *SAS-Cloud* (see Section 6) shows that it can protect all the possible attacks. Therefore, *SAS-Cloud* is more secure than other schemes.

Table 3 is given to show the storage cost, computational cost and communication overhead comparison of the schemes in [13–16, 23, 32, 34, 36, 37] with the proposed *SAS-Cloud*. Here, only login and authentication phases have been considered due to rapid and maximum use during online cloud services. On the other hand, registration of users is done offline and this phase is used only one time in the authentication systems. Therefore, communication and computational costs of registration phase can be neglected with respect to login and authentication phases. $T_{\mathcal{EXP}}$, $T_{\mathcal{H}}$, $T_{\mathcal{M}}$, $T_{\mathcal{ENC}}$ and $T_{\mathcal{DEC}}$ are the times required for exponentiation operation, hash operation, multiplication operation, symmetric key encryption and decryption respectively. However, it is well known that exponentiation operation takes more time than other operations and order of execution time can be expressed as: $T_{\mathcal{EXP}} >> T_{\mathcal{H}} \approx T_{\mathcal{ENC}} \ / \ T_{\mathcal{DEC}} > T_{\mathcal{M}}$ [17]. *SAS-Cloud* takes time for three exponentiation operation in two phases, which is the lower among ElGamal-based schemes in [23, 32, 34, 36, 37]. However, according to *MIRACL C/C++* Library with the specifications of system (i.e., processor: Intel(R) Core(TM) i5-4210U CPU @ 1.70GHz 2.40 GHz; RAM: 8 GB; 64-bit Windows 10; Visual C++ 2008 software), the time complexity of the different cryptographic operations is roughly calculated as follows: (1) $T_{\mathcal{ENC}} \ / \ T_{\mathcal{DEC}}$: For private key en/decryption, the time complexity is $\approx$ 0.1303 ms, (2) $T_{\mathcal{H}}$: For cryptographic hash function, the time complexity is $\approx$ 0.0004 ms, (3) $T_{\mathcal{EXP}}$: Time complexity for exponentiation is $\approx$ 1.8269 ms, and (4) $T_{\mathcal{M}}$: Time to execute multiplication operation is $\approx$ 0.0147 ms. According to the aforementioned information, a comparison graph (see Fig. 5) has been given as an evidence to show that *SAS-Cloud* takes less time to execute than related ElGamal cryptosystem based schemes. Here, we assume that the length of $\mathcal{ID}_i$ and $\mathcal{PW}_i$ are 64 bits each. Cryptographic hash function $\mathcal{H}(\cdot)$, threshold value δd, symmetric key encryp-

Table 2 Security attacks and functionality comparison of *SAS-Cloud* with related competing schemes

	Tan [13]	Yan et al. [14]	Mishra et al. [15]	Chuang and Chen [16]	Hwang and Li [23]	Shen et al. [32]	Yoon et al. [34]	Ramasamy and Muniyandi [36]	Lee et al. [37]	*SAS-Cloud*
SA^1	–	√	√	×	√	√	√	√	√	×
SA^2	×	×	√	×	√	√	√	√	×	×
SA^3	–	–	×	√	√	√	√	√	√	×
SA^4	–	√	×	√	√	√	√	√	√	×
SA^5	–	–	√	√	×	×	×	×	×	×
SA^6	√	–	×	√	–	–	×	–	×	×
SLS	×	×	√	√	×	×	√	×	√	√
SPS	×	×	√	√	×	×	√	×	√	√
MA	×	×	×	×	×	×	×	×	×	√
UT	√	√	√	×	√	√	√	√	√	×

SA^1: Password guessing attack, SA^2: Attack by insider, SA^3: Forge message attack, SA^4: Attack after stealing smart card/mobile device, SA^5: Replay attack, SA^6: DoS attack, SLS: Systematic login system, SPS: Systematic password change system, MA: Mutual authentication, UT: User traceability, ×: no, √: yes, –: Not applicable

Table 3 Communication, storage and computation costs comparison of *SAS-Cloud* with competing existing schemes

Schemes	***Cost of Storage*** (in bits)	***Cost of Communication*** (in bits) Login + Authentication	***Cost of Computation*** Login	Authentication
Tan [13]	384	576	$4T_{\mathcal{H}}$+1$T_{\mathcal{ENC}}$	$7T_{\mathcal{H}}$+1$T_{\mathcal{DEC}}$
Yan et al. [14]	640	960	$3T_{\mathcal{H}}$	$8T_{\mathcal{H}}$
Mishra et al. [15]	800	1120	$4T_{\mathcal{H}}$	$10T_{\mathcal{H}}$+1$T_{\mathcal{ENC}}$+1$T_{\mathcal{DEC}}$
Chuang and Chen [16]	576	960	$4T_{\mathcal{H}}$	$12T_{\mathcal{H}}$
Hwang and Li [23]	1152	2280	$3T_{\mathcal{EXP}}$+$T_{\mathcal{H}}$	$3T_{\mathcal{EXP}}$+$T_{\mathcal{H}}$
Shen et al. [32]	2176	2280	$3T_{\mathcal{EXP}}$+$T_{\mathcal{H}}$	$3T_{\mathcal{EXP}}$+$T_{\mathcal{H}}$
Yoon et al. [34]	4544	2496	$T_{\mathcal{EXP}}$+$2T_{\mathcal{H}}$+$T_{\mathcal{M}}$	$3T_{\mathcal{EXP}}$+$4T_{\mathcal{H}}$+$4T_{\mathcal{M}}$
Ramasamy and Muniyandi [36]	1152	3432	$2T_{\mathcal{EXP}}$+$T_{\mathcal{H}}$	$3T_{\mathcal{EXP}}$+$T_{\mathcal{H}}$
Lee et al. [37]	384	3456	$3T_{\mathcal{EXP}}$+$T_{\mathcal{H}}$+$T_{\mathcal{M}}$	$4T_{\mathcal{EXP}}$+$8T_{\mathcal{H}}$
SAS-Cloud	640	2432	$2T_{\mathcal{EXP}}$+$5T_{\mathcal{H}}$+$T_{\mathcal{M}}$	$T_{\mathcal{EXP}}$+$7T_{\mathcal{H}}$+$T_{\mathcal{M}}$

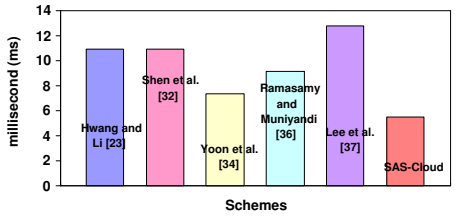

Fig. 5: Execution time in login and authentication phases of various related schemes (ElGamal cryptosystem-based): a comparison with *SAS-Cloud*

tion/decryption, random numbers, and timestamp return 128 bits each. Exponentiation operation provides 1024 bits. The communication cost of the proposed *SAS-Cloud* for login message is $(1024 + 1024 + 128)$ = 2176 bits and message cerated in authentication and key agreement phase is $(128 + 128)$ = 256 bits. Therefore, overall communication cost is $(2176 + 256)$ = 2432 bits, which is less than related ElGamal-based schemes in [34, 36, 37]. The storage cost of *SAS-Cloud* is $(128 + 128 + 128 + 128)$ = 512 bits, which is also less than related schemes in [14–16, 23, 32, 34, 36].

After resisting all possible attacks (based on random oracle model) as discussed in Section 6, *SAS-Cloud* has better trade-off among communication, computational and storage costs compared to the existing related schemes.

8 Satisfaction to Use of *SAS-Cloud*

1. *Efficient login system*: By some unwanted mistakes, if a user $\mathcal{U}_i$ inputs a faulty password as well as faulty identity in login phase of *SAS-Cloud*, the mobile device can identify the incorrect inputs before going to create a login message. This is because, the mobile calculates $\mathcal{B}_i' = \widehat{\mathcal{B}_i} \oplus \mathcal{H}(\mathcal{ID}_i \| \mathcal{PW}_i)$ and verifies $des(\mathcal{B}_i^*, \mathcal{B}_i') \leq \delta d$. For the incorrectness, the mobile discards $\mathcal{U}_i$; otherwise, it computes $\theta_i' = \widehat{\theta_i} \oplus \mathcal{H}(\mathcal{ID}_i \| \mathcal{PW}_i)$, $\phi_i' \leftarrow Rep(\mathcal{B}_i^*, \theta_i')$, $\mathcal{PWR}_i' = \mathcal{H}(\mathcal{PW}_i \| \phi_i')$, $A_i' = C_i \oplus \mathcal{PWR}_i'$, $D_i' = \mathcal{H}(A_i')$ and compares $D_i' =? D_i$. For the dissimilar result, the mobile discards $\mathcal{U}_i$; otherwise, considers $\mathcal{PW}_i$ and $\mathcal{ID}_i$ as correct inputs. Therefore, in *SAS-Cloud* for the wrong inputs, no login message will be generated which reduces extra communication overhead.

2. *Efficient password update system*: By some unwanted mistakes, if a user $\mathcal{U}_i$ inputs a faulty password as well as faulty identity in login phase of *SAS-Cloud*, the mobile device can identify the incorrect inputs before giving licence to the users to select their new password. This is because, the mobile follows the same steps as mentioned above to verify the correctness of entered inputs. Only for the correct inputs, the mobile gives licence to $\mathcal{U}_i$ to select new password. On the other hand, to update the password of a user, there is no need any communication between the mobile device and the

service provider. Hence, overhead for the communications is also decrease in *SAS-Cloud*.

3. *Identify of Denial of Service (DoS) attack*: Before going to create a login message, mobile device checks the password, biometric and identity of a user $\mathcal{U}_i$ in *SAS-Cloud*. That means, for any wrong input from user, mobile device does not generate login message. Now, if a login request message of $\mathcal{U}_i$ is discarded by service provider $\mathcal{SP}$, then it can be said that the login message has been tampered with another party (i.e., attacker) or corrupted by some reasons. Therefore, the user may take necessary action to stop DoS attack by informing service provider.

4. *Satisfaction of mutual authentication*: In *SAS-Cloud*, the service provider $\mathcal{SP}$ calculates and agrees on a secret session key SK_i after checking the authenticity of the user $\mathcal{U}_i$ through login message and after that, $\mathcal{SP}$ transmits a reply message to $\mathcal{U}_i$. Similarly, $\mathcal{U}_i$ goes for the same secret session key SK_i with $\mathcal{SP}$ after checking the authenticity of $\mathcal{SP}$ through reply message. Hence, two-way verification has been done in *SAS-Cloud*. Beside this, *SAS-Cloud* can protect all the possible security attacks (see, Section 6). Hence, *SAS-Cloud* satisfies mutual authentication.

5. *Untraceability of user*: In *SAS-Cloud*, identity of a user $\mathcal{U}_i$ is dynamic for every session. This is because, DID_i is computed as $(\mathcal{ID}_i \| r_i) \cdot E_i \bmod q$, where $E_i = PK^{r_i} \bmod q$ and r_i is a random number, which will be non-identical for each session in random fashion. Thus for different sessions, DID_i will be changed and as security analysis of *SAS-Cloud* (see Theorem 1) shows that $\mathcal{ID}_i$ cannot be extracted from known parameter for an adversary. Therefore, it can be claimed that the adversary cannot trace the user, which means the adversary is unable to locate the valid user's existence.

9 Conclusion

This work observed that most of the authentication protocols using hash function and ElGamal cryptosystem for cloud based applications are affected by security attacks and are unable to hide the actual identities of the end users during login session. Therefore, this work has introduced a secure ElGamal-based authentication scheme called *SAS-Cloud*. Analysis of security of *SAS-Cloud* using random oracle model shows that it is secure from all possible attacks. Performance comparison of *SAS-Cloud* with competing schemes has shown that the proposed scheme is more efficient than these competing schemes. In addition, as biometric features like finger print, iris scan, retina scan, and hand geometry are used with password in *SAS-Cloud*, therefore they can improve the security label of password-based authentication scheme. In future, this work will be extended to provide secure authentication among cloud server and Internet enabled devices so that a complete security framework can be build for IoT applications.

10 References

1 Zanella, A., Bui, N., Castellani, A., Vangelista, L., Zorzi, M.: 'Internet of things for smart cities', *IEEE Internet of Things Journal*, 2014, **1**, (1), pp. 22–32

2 Mell, P.M., Grance, T. 'The nist definition of cloud computing, computer security division, information technology laboratory, national institute of standards and technology gaithersburg'. (, 2011.

3 Lamport, L.: 'Password authentication with insecure communication', *Commun ACM*, 1981, **24**, (11), pp. 770–772

4 Li, X., Niu, J., Khan, M.K., Liao, J.: 'An enhanced smart card based remote user password authentication scheme', *Journal of Network and Computer Applications*, 2013, **36**, (5), pp. 1365–1371

5 Maitra, T., Amin, R., Giri, D., Srivastava, P.D.: 'An efficient and robust user authentication scheme for hierarchical wireless sensor networks without tamper-proof smart card', *I J Network Security*, 2016, **18**, (3), pp. 553–564

6 Mir, O., v. d. Weide, T., Lee, C.C.: 'A secure user anonymity and authentication scheme using avispa for telecare medical information systems', *Journal of Medical Systems*, 2015, **39**, (9)

7 Guo, C., Chang, C.C.: 'Chaotic maps-based password-authenticated key agreement using smart cards', *Communications in Nonlinear Science and Numerical Simulation*, 2013, **18**, (6), pp. 1433–1440

8 Maitra, T., Islam, S.H., Amin, R., Giri, D., Khan, M.K., Kumar, N.: 'An enhanced multi-server authentication protocol using password and smart-card: cryptanalysis and design', *Security and Communication Networks*, 2016, **9**, (17), pp. 4615–4638

9 Odelu, V., Das, A.K., Wazid, M., Conti, M.: 'Provably secure authenticated key agreement scheme for smart grid', *IEEE Transactions on Smart Grid*, 2018, **9**, (3), pp. 1900–1910

10 Jan, M.A., Khan, F., Alam, M., Usman, M.: 'A payload-based mutual authentication scheme for internet of things', *Future Generation Computer Systems*, 2019, **92**, pp. 1028–1039

11 Jain, A.K., Ross, A., Pankanti, S.: 'Biometrics: a tool for information security', *IEEE Transactions on Information Forensics and Security*, 2006, **1**, (2), pp. 125–143

12 Giri, D., Sherratt, R.S., Maitra, T.: 'A novel and efficient session spanning biometric and password based three-factor authentication protocol for consumer usb mass storage devices', *IEEE Transactions on Consumer Electronics*, 2016, **62**, (3), pp. 283–291

13 Tan, Z.: 'An efficient biometrics-based authentication scheme for telecare medicine information systems', *Przeglad Elektrotechniczny*, 2013, pp. 200–204

14 Yan, X., Li, W., Li, P., Wang, J., Hao, X., Gong, P.: 'A secure biometrics-based authentication scheme for telecare medicine information systems', *Journal of Medical Systems*, 2013, **37**, (5)

15 Mishra, D., Mukhopadhyay, S., Chaturvedi, A., Kumari, S., Khan, M.K.: 'Cryptanalysis and improvement of yan et al.'s biometric-based authentication scheme for telecare medicine information systems', *Journal of Medical Systems*, 2014, **38**, (6)

16 Chuang, M.C., Chen, M.C.: 'An anonymous multi-server authenticated key agreement scheme based on trust computing using smart cards and biometrics', *Expert Systems with Applications*, 2014, **41**, (4, Part 1), pp. 1411–1418

17 Maitra, T., Giri, D.: 'An efficient biometric and password-based remote user authentication using smart card for telecare medical information systems in multi-server environment', *Journal of Medical Systems*, 2014, **38**, (12)

18 Islam, S.H., Vijayakumar, P., Bhuiyan, M.Z.A., Amin, R., M., V.R., Balusamy, B.: 'A provably secure three-factor session initiation protocol for multimedia big data communications', *IEEE Internet of Things Journal*, 2017, **PP**, (99), pp. 1–11

19 Wazid, M., Das, A.K., Kumari, S., Li, X., Wu, F.: 'Provably secure biometric-based user authentication and key agreement scheme in cloud computing', *Security and Communication Networks*, 2016, **9**, (17), pp. 4103–4119

20 Giri, D., Sherratt, R.S., Maitra, T., Amin, R.: 'Efficient biometric and password based mutual authentication for consumer usb mass storage devices', *IEEE Transactions on Consumer Electronics*, 2015, **61**, (4), pp. 491–499

21 Giri, D., Maitra, T., Amin, R., Srivastava, P.D.: 'An efficient and robust rsa-based remote user authentication for telecare medical information systems', *Journal of Medical Systems*, 2014, **39**, (1)

22 Amin, R., Maitra, T., Giri, D., Srivastava, P.D.: 'Cryptanalysis and improvement of an rsa based remote user authentication scheme using smart card', *Wireless Personal Communications*, 2017, **96**, (3), pp. 4629–4659

23 Hwang, M.S., Li, L.H.: 'A new remote user authentication scheme using smart cards', *IEEE Transactions on Consumer Electronics*, 2000, **46**, (1), pp. 28–30

24 Amin, R., Biswas, G.P. In: 'Remote access control mechanism using rabin public key cryptosystem'. (New Delhi: Springer India, 2015. pp. 525–533

25 Maitra, T., Obaidat, M.S., Islam, S.H., Giri, D., Amin, R.: 'Security analysis and design of an efficient ecc-based two-factor password authentication scheme', *Security and Communication Networks*, 2016, **9**, (17), pp. 4166–4181

26 Han, L., Tan, X., Wang, S., Liang, X.: 'An efficient and secure three-factor based authenticated key exchange scheme using elliptic curve cryptosystems', *Peer-to-Peer Networking and Applications*, 2018, **11**, (1), pp. 63–73

27 Odelu, V., Das, A.K., Goswami, A.: 'A secure biometrics-based multi-server authentication protocol using smart cards', *IEEE Transactions on Information Forensics and Security*, 2015, **10**, (9), pp. 1953–1966

28 He, D., Zeadally, S., Kumar, N., Wu, W.: 'Efficient and anonymous mobile user authentication protocol using self-certified public key cryptography for multi-server architectures', *IEEE Transactions on Information Forensics and Security*, 2016, **11**, (9), pp. 2052–2064

29 Jiang, Q., Ma, J., Wei, F.: 'On the security of a privacy-aware authentication scheme for distributed mobile cloud computing services', *IEEE Systems Journal*, 2018, **12**, (2), pp. 2039–2042

30 ElGamal, T. In: 'A public key cryptosystem and a signature scheme based on discrete logarithms'. (Berlin, Heidelberg: Springer Berlin Heidelberg, 1985. pp. 10–18

31 Chan, C.K., Cheng, L.M.: 'Cryptanalysis of a remote user authentication scheme using smart cards', *IEEE Transactions on Consumer Electronics*, 2000, **46**, (4), pp. 992–993

32 Shen, J.J., Lin, C.W., Hwang, M.S.: 'A modified remote user authentication scheme using smart cards', *IEEE Transactions on Consumer Electronics*, 2003, **49**, (2), pp. 414–416

33 Leung, K.C., Cheng, L.M., Fong, A.S., Chan, C.K.: 'Cryptanalysis of a modified remote user authentication scheme using smart cards', *IEEE Transactions on Consumer Electronics*, 2003, **49**, (4), pp. 1243–1245

34 Yoon, E.J., Ryu, E.K., Yoo, K.Y.: 'Efficient remote user authentication scheme based on generalized elgamal signature scheme', *IEEE Transactions on Consumer Electronics*, 2004, **50**, (2), pp. 568–570

35 Tian, X., Zhu, R.W., Wong, D.S.: 'Improved efficient remote user authentication schemes', *International Journal of Network Security*, 2007, **4**, (2), pp. 149–154

36 Ramasamy, R., Muniyandi, A.P.: 'New remote mutual authentication scheme using smart cards.', *Transactions on Data Privacy*, 2009, **2**, (2), pp. 141–152

37 Lee, Y.C., Hsieh, Y.C., Lee, P.J., You, P.S.: 'Improvement of the elgamal based remote authentication scheme using smart cards', *Journal of Applied Research and Technology*, 2014, **12**, (6), pp. 1063–1072

38 Maitra, T., Obaidat, M.S., Amin, R., Islam, S.H., Chaudhry, S.A., Giri, D.: 'A robust elgamal-based password-authentication protocol using smart card for client-server communication', *International Journal of Communication System*, 2016, **30**, (11), pp. 1–12

39 Henry, E.R.: 'Classification and Uses of Finger Prints'. (George Routledge and Sons, London, 1900)

40 Obaidat, M., Boudriga, N.: 'Security of e-Systems and Computer Networks'. (New York, NY, USA: Cambridge University Press, 2007)
41 Dolev, D., Yao, A.C.: 'On the security of public key protocols', *Information Theory, IEEE Transactions on*, 1983, **29**, (2), pp. 198–208
42 Messerges, T.S., Dabbish, E.A., Sloan, R.H.: 'Examining smart-card security under the threat of power analysis attacks', *IEEE Trans Comput*, 2002, **51**, (5), pp. 541–552